DOCUMENT RESUME

ED 059 596 EM 009 588

AUTHOR	Tschudi, Ronald R.; Meredith, Joseph C.
TITLE	The "PROBE" Retrieval Program; A Description.
INSTITUTION	Governors State Univ., Park Forest South, Ill.; Indiana Univ., Bloomington. Research Computing Center.
PUB DATE	10 Feb 72
NOTE	11p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Programs; Computers; Educational Technology; Information Dissemination; *Information Retrieval; Information Science; *Information Services; Information Utilization
IDENTIFIERS	ERIC; *PROBE

ABSTRACT

PROBE is a search and retrieval program designed for use with large tape files of bibliographic data such as the ERIC system's "Research in Education" and "Current Index to Journals in Education" data bases. The program, implemented on a CDC 6600 computer and being developed for the IBM 360, is characterized by a broad range of capabilities and options which are described in this report. The general capabilities of PROBE are briefly described, and further sections discuss PROBE search procedures and specification. Two separete levels of search parameter specification--SPEC and QUERY--are described in more detail. A brief section on the application of the PROBE program concludes the report. (SH)

FILMED FROM BEST AVAILABLE COPY

ED 059596

THE "PROBE" RETRIEVAL PROGRAM - A DESCRIPTION

by

Ronald R. Tschudi
Research Computing Center
Indiana University
Bloomington, Indiana 47401

and

Joseph C. Meredith
Learning Resources Center
Governors State University
Park Forest South, Illinois 60466

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

EM 009 588

February 10, 1972

TECHNICAL REPORT ----------

THE "PROBE" RETRIEVAL PROGRAM - A DESCRIPTION

Abstract:

This report describes the "PROBE" search and retrieval program in terms of its current application to ERIC tapes, the scope of its service options, and the forms in which search parameters are expressed for processing in the system.

THE "PROBE" RETRIEVAL PROGRAM - A DESCRIPTION

Ronald R. Tschudi

Joseph C. Meredith

"PROBE" is a search and retrieval program designed for use with large tape files of bibliographic data such as the ERIC Research in Education (RIE) and Current Index to Journals in Education (CIJE) data bases. It is implemented on a CDC 6600 computer at Indiana University, to provide an information service based on the ERIC tapes, which are routinely converted and combined for the purpose. A second version for the IBM 360 is under development.

The program is characterized by an unusually broad range of capabilities and options, of which the user needs to be aware in order to employ it to the best advantage. These are described in the following report.

GENERAL

When invoked, PROBE causes the magnetic tape file to be searched according to sets of parameters supplied by one or more users. The program then causes all retrieved records to be segregated according to the particular sets that they

satisfy, and then to be printed in full or in part, depending on pre-set limits. In addition, the program returns a report of the number of hits obtained, and a report of the number of records printed, for each set.

PROCEDURE

A PROBE search is initiated as follows, using punched cards inserted in a small deck that invokes the master program:

First, default parameters are set by means of statements covering certain standard options, such as (1) the maximum number of retrieved records to be printed in full, for any one query (except as modified by a local query parameter); (2) the specific fields to be included in a "full" printout; (3) the range of fields to be searched in order to meet the combined requirement of all searches in the run; and (4) the symbol to be used as a string delimiter...normally an asterisk.

Next, cards for the individual sets of search parameters are supplied. Each set can be thought of as a separate program. That is, it is individually "compiled" by PROBE, provided it contains no syntactical error. Otherwise, though it will fail to be compiled, the failure is localized to that one set, and the cause thereof will be stated in the ultimate output.

SPECIFICATION

The way in which searches are specified permits great flexibility and scope as far as the user is concerned, while affording maximum efficiency and economy in program execution.

For each set of search parameters, two separate levels of specification are used - "SPEC" and "QUERY". At least one numbered specimen of each must be present in any search. In effect, the program breaks the decision process into two parts: "Is a certain element found?" and "What is to be done about found elements?"

"SPEC"

"SPEC" statements are used to establish conditions of either "true" or "false" for

one or more specified strings

each composed of

one or more characters and/or blanks

in respect to

one or more specified fields.

For example:

SPEC(1) FIELD(35) *ADMINISTR*

creates a requirement (SPEC number 1) that the program search field number 35 of every record for the string of characters '<u>administr</u>' in that order, with no intervening characters or blanks. If found, as with <u>administrate</u>, <u>administration</u>, <u>administrative</u>, <u>administrator</u>, (but not <u>administer</u> or

or administered) a "truth condition" for that particular specification is set, and is linked to the record containing that string.

The SPEC statement may combine two or more string requirements, e.g.:

SPEC(2) FIELD(44) *WINTER* *SPRING*

Meaning: Search field 44 (the abstract field, in the ERIC tapes) for winter, followed (at any interval) by spring.

SPEC(3) FIELD(44) *WINTER* AND *SPRING*

Meaning: Search the abstract field for winter and spring, and set the truth condition if both are found, regardless of their order.

SPEC(4) FIELD(44) *WINTER* OR *SPRING*

Meaning: Search the abstract field for winter. If found, set the truth condition for that statement and proceed to the next statement. If winter is not found, search the field for spring for final determination of "true" or "false." (Note that economy of search time is furthered by putting the most likely string first, in an "or"ed SPEC. With an "and"ed SPEC, putting the least likely string first is more efficient.)

SPEC(5) FIELD(35,44) * COGNITION*

Meaning: Search both the descriptor field and the abstract field for 'cognition'. (Note that by specifying a leading blank - '_COGNITION' - we have ruled out recognition.)

SPEC(5) FIELD(35,44) * COGNITION* AND *RETARDED*

the same, but both strings must occur in either one or the

other of the specified fields, not split between them.

```
SPEC(6) FIELD(36,44) *ALABAMA* OR *ARKANSAS*  )
                                              )
OR *TEXAS* OR *LOUISIANA* OR *FLOR.           )   80 cols.
IDA* OR *MISSISSIPPI*                             (continuation)
```

Meaning: Search first the identifier field for Alabama. If successful, set the truth condition for SPEC(6) and terminate that particular SPEC-search on the record at hand. If not, continue. If none of the strings is found in field 36, repeat process on field 44. (Note that overflow to a continuation card is signalled by a period in column 80.)

There is no limit on the number of strings and/or fields that can be combined in a single SPEC. However, only one type of logical operator (AND,OR) may be used in the same SPEC. Combinations such as *OATS* AND (*PEAS* OR *BARLEY*) are not processed at the SPEC level.

SPEC can also be used to test numerical values. For example, the following statement would cause the field containing the ERIC accession number of each record to be examined, setting the truth condition for records having ED-numbers greater than ED32000:

```
SPEC(7) FIELD(16) GT32000
```

This feature allows the user to limit his search to a particular segment of the file.

There is no limit on the number of SPECs that may be assembled for use in a particular search set.

"QUERY"

A "QUERY" statement establishes whether a given combination of SPECs is "true" or "false", by virtue of each of its components being (logically) true or false. Statements at the QUERY level take the following generalized form:

QUERY(N) A operator B operator C .....

where N is an arbitrarily assigned number which identifies the results in the printout; A, B, C, etc... are numbers which refer to previously stated SPECs; and operators are AND, OR, or NOT. For example:

QUERY(1) 1 AND 2

Meaning: Establish whether SPEC(1) and SPEC(2), when applied to a particular record, have both been established as "true." In every case where this is so, designate the record as a hit, to be duly printed with the identification "Query 1."

Query(2) 3 OR 4

Meaning: Establish whether either SPEC(3) or SPEC(4) has been set as "true", for a particular record. If so, the QUERY succeeds, and the record will be retrieved and printed.

QUERY(3) 5 AND NOT 6

Meaning: The QUERY will succeed if SPEC(5) is true, except in cases where SPEC(6) is also true.

QUERY(5) (1 AND 2) OR (7 OR (3 AND NOT (4 OR (5 AND 6))))

Meaning: The record will be tested for a Boolean combination of SPECs. Blanks are immaterial; the foregoing might have been

stated:

QUERY(5)(1AND2)OR(7OR(3ANDNOT(4OR(5AND6))))

Other variables may be added, for example the "MAX" function:

QUERY(6) MAX(250) 1

Meaning: Retrieve all documents for which SPEC(1) is true, and record the total number retrieved, but print only the first 250 of them in full. The remainder are simply to be listed according to number, author, and title. "MAX" in a QUERY statement overrides the default maximum set in the initial decklet.

Weighted searches may also be carried out by adding special TOTAL and WEIGHT elements to any QUERY. The desired threshold is indicated with the control word "TOTAL", followed by the "WEIGHT" to be equalled or exceeded in order to constitute a hit. The "argument" following a WEIGHT code is in the form of three numbers separated by commas, that we might call"SPEC-weight triplets." Each "triplet" names a SPEC, indicates its "true" value, and indicates its "false" value (usually zero). For example:

WEIGHT(1,5,0)

SPEC no. true value false value

Thus...

QUERY(1) MAX(100) TOTAL(5) WEIGHT(1,4,0,2,3,0,3,3,0)

succeeds, for a particular record, if at least two out of the three weighted SPECs are true.

Just as there is no limitation on the number of SPECs that may be used in a search set, there is none on the number of QUERYs that may - in a decklet covering a particular search - refer to one or more of them in various combinations. SPECs and QUERYs may be interspersed, as long as the SPECs referred to in a particular QUERY precede it. Thus sequence A would succeed, whereas B would fail:

A	B
SPEC(1)	SPEC(1)
SPEC(2)	SPEC(2)
SPEC(3)	SPEC(3)
QUERY(1) 1 AND 2 AND 3	QUERY(1) 1 AND 4
SPEC(4)	SPEC(4)
QUERY(2) 1 AND 2 AND 4	QUERY(2) 3

The rules of syntax are simple and few: SPEC, FIELD, QUERY, MAX, TOTAL, and WEIGHT commands must follow the form given in the above examples; unbalanced parentheses are illegal; unbalanced string delimiters are illegal; a period in column 80 must precede a continuation card; and comment cards must always begin with a "C".

In other respects, the user enjoys virtual free-formatting: statements may begin in any column except 80; comments may begin in any column but 1; and blanks may be freely used or completely omitted between any and all elements of any statement, except of course in character strings where their presence or absence is a specified condition.

APPLICATION

Although the PROBE program has been used to date chiefly in conducting searches of the ERIC tapes, it can readily be adjusted to deal with similarly structured data, and has in fact been successfully tested against MARC II tapes. Aside from typical search services, it has also been found useful as a means of splitting out large subsets of records on the basis of originating Clearinghouses, academic diciplines, and the like.

This technical report does not address the subject of search strategies, since these entail detailed consideration of the nature of the file, the indexing terms used and the evolution of their use, the relative speeds of limited field search vs. full text search, and the effectiveness of the query negotiation interface.

#